Computers and Fluids 138 (2016) 38–50

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

A balanced-force control volume finite element method for interfacial flows with surface tension using adaptive anisotropic unstructured meshes

Zhihua Xie[a,b,*], Dimitrios Pavlidis[a], Pablo Salinas[a], James R. Percival[a], Christopher C. Pain[a], Omar K. Matar[b]

[a] Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, London, UK
[b] Department of Chemical Engineering, Imperial College London, London, UK

ARTICLE INFO

Article history:
Received 30 January 2016
Revised 28 July 2016
Accepted 12 August 2016
Available online 13 August 2016

Keywords:
Curvatures
Discontinuous Galerkin
Multiphase flows
Surface tension
Unstructured mesh
Volume of fluid method

ABSTRACT

A balanced-force control volume finite element method is presented for three-dimensional interfacial flows with surface tension on adaptive anisotropic unstructured meshes. A new balanced-force algorithm for the continuum surface tension model on unstructured meshes is proposed within an interface capturing framework based on the volume of fluid method, which ensures that the surface tension force and the resulting pressure gradient are exactly balanced. Two approaches are developed for accurate curvature approximation based on the volume fraction on unstructured meshes. The numerical framework also features an anisotropic adaptive mesh algorithm, which can modify unstructured meshes to better represent the underlying physics of interfacial problems and reduce computational effort without sacrificing accuracy. The numerical framework is validated with several benchmark problems for interface advection, surface tension test for equilibrium droplet, and dynamic fluid flow problems (fluid films, bubbles and droplets) in two and three dimensions.

1. Introduction

Interfacial flows with surface tension appear in many engineering applications, e.g. micro-fluidics, oil-and-gas transportation systems, geophysical flows and nuclear reactors. These applications typically involve the motion of bubbles, droplets, fluid films and jets, featuring tremendous complexity in interfacial topology and a large range of spatial scales.

A key requirement for modelling interfacial flows is a method for tracking or capturing the interface [1]. Numerous methods have been proposed and used to simulate interfacial flows on a fixed mesh, such as marker-and-cell [2], volume-of-fluid (VOF) [1,3,4], front-tracking [5], level set [6,7], phase field [8] and particle [9] methods. In particular, VOF methods are widely used due to the inherent properties of: mass conservation, computational efficiency and easy implementation. From a general point of view, there are two classes of algorithms to solve the transport equation of volume fraction: geometric and algebraic computation [4]. In the geometric VOF methods [1], interfaces are first reconstructed from the volume fraction data so that a geometric profile is found which approximates the actual interface location. Then changes in volume fraction are calculated by integrating volume fluxes across cell boundaries, using flux splitting or unsplitting schemes. In the algebraic computation [10,11], the interface is captured by solving the transport equation of volume fraction with a differencing scheme without reconstructing the interface, such as the flux-corrected transport scheme [10] and using the normalised variable diagram (NVD) [12] concept to switch between different differencing schemes [11].

As the dynamics of interfacial flows are highly unsteady and the shape and location of the interface are changing during the simulation, interface calculation methods based on a fixed mesh need finer mesh resolution in order to capture the details, which will significantly increase computational efforts. The alternative is to consider the use of dynamically adaptive mesh methods, where the mesh resolution can vary in time in response to the evolving solution fields. For example, a finer mesh could be placed around the interface during its development while a coarser mesh could be used away from the interface while the flow is less dynamic. There are some examples of the use of adaptive mesh refinement for structured meshes with volume of fluid [13] and

* Corresponding author.
E-mail address: zhihua.xie@hotmail.com, z.xie@imperial.ac.uk (Z. Xie).
URL: http://amcg.ese.imperial.ac.uk. (Z. Xie, D. Pavlidis, P. Salinas, J.R. Percival, C.C. Pain)

http://dx.doi.org/10.1016/j.compfluid.2016.08.005

hybrid level set/front tracking [14] methods. Unstructured meshes are very attractive to deal with complex geometries in engineering applications and there is an example of adaptive unstructured meshes with the level set method [15]. Recently, a novel algebraic VOF interface capturing method based on a compressive advection method on adaptive unstructured meshes has been developed by [16] and some examples of its application for multiphase flows in two dimensions can be found in [17,18]. It is also worth mentioning that many computational domains have a high aspect ratio and most interfacial flow phenomena can possess strong anisotropies, therefore anisotropic mesh resolution may be required to optimally represent the dynamics of the flow. Some examples of anisotropic unstructured mesh adaptivity can be found in [19].

In this paper, we focus on the surface tension force model in three dimensions. Many different types of surface tension force model have been proposed in the past, where the continuum surface force (CSF) method [20] has been widely used in the level set and volume of fluid methods. The level set function is a smooth continuous function, which can estimate accurately the curvature for the surface tension, however the standard level set method might suffer from the mass conservation. The volume of fluid method is mass conservative, however it is difficult to calculate the surface tension force accurately due to the step function of the volume fraction. This leads to the development of coupled level set and volume of fluid (CLSVOF) method [21], which takes advantage of both methodologies. Recently the balanced-force algorithm for surface tension model has become popular in structured Cartesian grids due to the use of a height function for curvature calculation in the volume of fluid method [22] and the level set function in the CLSVOF method [23]. It has also been extended for adaptive mesh refinement for structured Cartesian grids [13]. However, less attention has been paid to the balanced-force algorithm for fully unstructured meshes, even without mesh adaptivity.

The motivation for this work is to develop a balanced-force control volume finite element method for three-dimensional interfacial flows with surface tension on adaptive anisotropic unstructured meshes, which can modify unstructured meshes to better represent the underlying physics of interfacial problems and reduce computational efforts without sacrificing accuracy. A new balanced-force algorithm for the CSF model is proposed within the interface capturing framework based on the volume of fluid method.

The remainder of this paper is organised as follows. Description of the model and numerical methods is given in Section 2. Numerical examples of pure advection, static drop in equilibrium, fluid films, bubbles and droplets are presented in Section 3. Finally, some concluding remarks and future work are given in Section 4.

2. Mathematical model and numerical methods

In this section, we first describe the mathematical model and then we present our numerical framework based on the control volume and finite element method. The new balanced-force algorithm for the CSF model is proposed within the interface capturing framework and discussed in detail.

2.1. Governing equations

In multi-component flows, a number of components exist in one or more phases (one phase is assumed here but is easily generalised to an arbitrary number of phases or fluids). Let α_i be the mass fraction of component i, where $i = 1, 2, .., N_c$ and N_c denotes the number of components. The density and dynamic viscosity of component i are ρ_i and μ_i, respectively. A constraint on the system is:

$$\sum_{i=1}^{N_c} \alpha_i = 1. \tag{1}$$

For each fluid component i, the conservation of mass may be defined as,

$$\frac{\partial}{\partial t}(\alpha_i) + \nabla \cdot (\alpha_i \mathbf{u}) = 0, \quad i = 1, 2, \ldots, N_c, \tag{2}$$

and the equations of motion of an incompressible fluid may be written as:

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot [\mu (\nabla \mathbf{u} + \nabla^{\mathrm{T}} \mathbf{u})] + \rho \mathbf{g} + \mathbf{F}_\sigma, \tag{3}$$

where t is the time, $\mathbf{u}$ is velocity vector, p is the pressure, the bulk density is $\rho = \sum_{i=1}^{N_c} \alpha_i \rho_i$, the bulk dynamic viscosity is $\mu = \sum_{i=1}^{N_c} \alpha_i \mu_i$, $\mathbf{g}$ is the gravitational acceleration vector, and $\mathbf{F}_\sigma$ is the surface tension force. In the present study, we focus on the surface tension model for interfacial flows with two components, i.e. $N_c = 2$.

2.2. Numerical methods

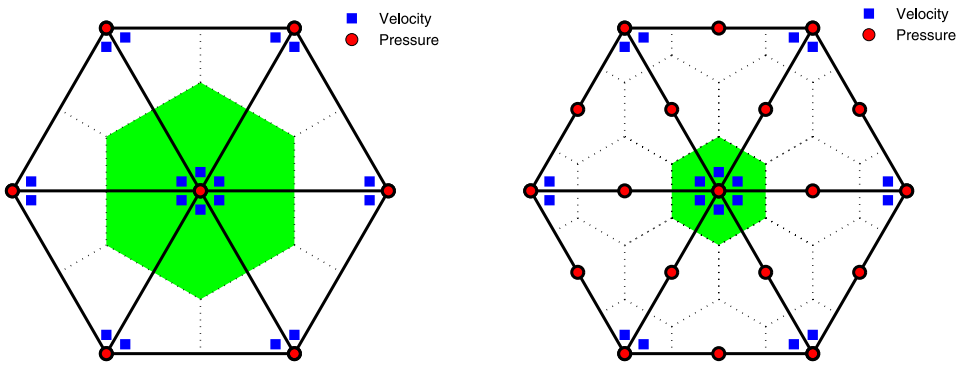
There are several numerical discretisation methods that solve the governing equations, such as the finite difference method, finite volume method and finite element method [24]. The finite element method with unstructured meshes is very attractive, as it provides accuracy and great flexibility in dealing with complex geometries and moving interfaces. In addition, with the finite element method it is possible to develop a compact high-order discretisation by applying higher-order polynomial expansions within every element.

2.2.1. Computational grid

The numerical framework consists of control volume and finite element formulation and also a discontinuous/continuous finite element pair. In the formulation, the domain is discretised into triangular or tetrahedral elements and in this work, they are either $P_1DG\text{-}P_1$ elements (linear discontinuous velocity between elements and linear continuous pressure between elements) or $P_1DG\text{-}P_2$ elements (linear discontinuous velocity between elements and quadratic continuous pressure between elements) [16]. Fig. 1 shows the locations of the degrees of freedom for the $P_1DG\text{-}P_1$ and $P_1DG\text{-}P_2$ elements and the boundaries of the control volumes in two dimension (2D).

2.2.2. Temporal discretisation

Time stepping schemes include first-order schemes, such as the explicit forward Euler and implicit backward Euler schemes. The explicit scheme is more easy and straightforward to implement but imposes restriction on the time step size due to the Courant–Friedrichs–Lewy (CFL) condition, whereas the implicit scheme is stable for large Courant numbers but is more dissipative. A new time discretisation scheme is employed here. When high-order discretisation is sought, the method is based on traditional Crank–Nicolson time stepping. The Crank–Nicolson method is often used because it has the simplicity of a two-level time stepping method, is unconditionally stable and second-order accurate. However, for interface-capturing applications, the time discretisation scheme is based on the explicit forward Euler time stepping method. This introduces negative dissipation and is thus a compressive scheme which helps maintain sharp interfaces. The use of time steps of the order of the grid Courant number and above can result in numerical oscillations and unphysical solutions. For this reason an adaptive θ parameter is introduced [16] and shown explicity in Section 2.2.3, in which the forward Euler time stepping method is obtained for $\theta = 0$, the Crank–Nicolson method is obtained for $\theta = 0.5$ and the backward Euler method is obtained for $\theta = 1$.



(a) P_1DG-P_1 element (b) P_1DG-P_2 element

Fig. 1. (a) Finite element used to discretise the governing equations. The central position of key solution variables (velocity and pressure) are indicated here for the P_1DG-P_1 (a) and P_1DG-P_2 (b) element pairs in 2D. Diagram also shows the relationship between intersecting control volumes (shaded area with dotted line boundary) and elements (solid line boundary) for the P_1 and P_2 elements.

2.2.3. Spatial discretisation for the global continuity and momentum equations

In order to discretise the above governing equations, a finite element representation for $\mathbf{u}$ and p is assumed, expressed in terms of their finite element basis functions $\mathbf{Q}_j$ and P_j, respectively, as:

$$\mathbf{u} = \sum_{j=1}^{\mathcal{N}_u} \mathbf{Q}_j \mathbf{u}_j \quad \text{and} \quad p = \sum_{j=1}^{\mathcal{N}_p} P_j p_j. \tag{4}$$

Here $\mathcal{N}_u$ and $\mathcal{N}_p$ are the total number of degrees of freedom for the velocity and pressure representations. $\mathbf{Q}_j = Q_j\mathbf{I}$, where $\mathbf{I}$ is the identical matrix.

When multiplying the global continuity equation with a quadratic continuous Galerkin (CG) basis function P_i and applying integration by parts once, the discrete form of the global continuity equation can be obtained as:

$$\int_V P_i\left(\nabla\cdot\mathbf{u}^{n+1}\right)dV \approx -\int_V (\nabla P_i)\cdot\mathbf{u}^{n+1}dV + \int_{\Gamma-\Gamma_{\rm bc}} P_i\mathbf{n}\cdot\mathbf{u}^{n+1}d\Gamma + \int_{\Gamma_{\rm bc}} P_i\mathbf{n}\cdot\mathbf{u}_{\rm bc}^{n+1}d\Gamma = 0, \tag{5}$$

where n is the current time level, $\mathbf{n}$ is the outward-pointing unit normal vector to the surface Γ of the domain V, and subscript bc means the value on the boundary.

When multiplying Eq. (3) with a linear discontinuous Galerkin (DG) basis function $\mathbf{Q}_i$ and applying integration by parts twice over each element e with the θ time stepping method, the discrete form of the momentum equation can be obtained as:

$$\begin{aligned}\int_{V_E}\mathbf{Q}_i\left(\frac{\partial(\rho\mathbf{u})}{\partial t}+\nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u})+\nabla p-\nabla\cdot[\mu(\nabla\mathbf{u}+\nabla^{\rm T}\mathbf{u})]-\rho\mathbf{g}-\mathbf{F}_\sigma\right)dV \\ \approx \int_{V_E}\mathbf{Q}_i\left(\frac{\rho(\mathbf{u}^{n+1}-\mathbf{u}^n)}{\Delta t}+\nabla p^{n+1}-\rho\mathbf{g}-\mathbf{F}_\sigma\right)dV \\ +\theta(\mathbf{S}_{i\rm ADV}^{n+1}+\mathbf{S}_{i\rm VIS}^{n+1})+(1-\theta)(\mathbf{S}_{i\rm ADV}^{n}+\mathbf{S}_{i\rm VIS}^{n}) \\ +\int_{\Gamma_{\rm bc}}\mathbf{Q}_i\mathbf{n}(p_{\rm bc}^{n+1}-p^{n+1})d\Gamma = 0,\end{aligned} \tag{6}$$

where $\mathbf{n}$ is the outward-pointing unit normal vector to the surface Γ_E of the element V_E, $\Gamma_{\rm bc}$ is a boundary with prescribed pressure, $\theta \in \{0, 1\}$ is the implicitness parameter, Δt is the time step size, $\mathbf{S}_{i\rm ADV}$ and $\mathbf{S}_{i\rm VIS}$ represent the discrete from for the advection and viscous terms, respectively.

The upwind discontinuous Galerkin method is used for the advection terms and the value for time level $n+1$ is:

$$\mathbf{S}_{i\rm ADV}^{n+1} = \int_{V_E}\mathbf{Q}_i\rho\mathbf{u}_*^{n+1}\nabla\cdot\mathbf{u}^{n+1}dV - \int_{\Gamma_E}\mathbf{Q}_i\rho\mathbf{u}_*^{n+1}\mathbf{n}\cdot(\mathbf{u}^{n+1}-\mathbf{u}_{\rm in}^{n+1})d\Gamma, \tag{7}$$

where $\mathbf{u}_{\rm in}$ is the upwind velocity calculated from the neighbouring element or boundary and the subscript $(_*)$ represents the latest value during the iteration in one time step. The advection term for the time level n can be obtained in a similar way.

For the viscous terms $\tau = \mu(\nabla\mathbf{u}+\nabla^{\rm T}\mathbf{u})$, we use a high order linear scheme which results in a compact stencil with an element coupling only to its surrounding elements. For example,

$$\mathbf{S}_{i\rm VIS}^{n+1} = \int_{V_E}\nabla\mathbf{Q}_i\cdot\tau^{n+1}dV - \frac{1}{2}\int_{\Gamma_E}\mathbf{Q}_i\mathbf{n}\cdot(\tau^{n+1}+\tau_{\rm nb}^{n+1})d\Gamma, \tag{8}$$

where $\tau_{\rm nb}$ is the value of τ in the neighbouring element along the face and the viscous term for the time level n can be obtained in a similar way.

In order to evaluate the viscous stress tensor τ on the boundary of element Γ_E in Eq. (8), we integrate over the volume of two neighbouring elements in order to calculate the derivatives on the element face between the two elements. For example, the derivative in x for the x component of $\mathbf{u}$ is obtained as:

$$\int_{V_{E1}} Q_i u_x dV = \int_{V_{E1}} Q_i\frac{\partial u}{\partial x}dV - \int_{\Gamma_{E1\cap E2}} Q_i n_x(u-u_{\rm nb})d\Gamma, \tag{9}$$

$$\int_{V_{E2}} Q_i u_x dV = \int_{V_{E2}} Q_i\frac{\partial u}{\partial x}dV - \int_{\Gamma_{E1\cap E2}} Q_i n_x(u-u_{\rm nb})d\Gamma, \tag{10}$$

in which $\Gamma_{E1\cap E2}$ is the shared face between element 1 ($E1$) and element 2 ($E2$), u is the x component of velocity $\mathbf{u}$ and $u_{\rm nb}$ is the value of u in the neighbouring element along the face $\Gamma_{E1\cap E2}$. It is worth noting that this is not only validated for discontinuous elements but also can be applied for continuous elements.

2.2.4. Projection method

The discretised form of the momentum (Eq. (6)) and global mass balance (Eq. (5)) equations are solved using a pressure projection method. This effectively eliminates the unknown velocity and solves a system of equations for pressure or pressure correction. The discretised momentum and continuity equations, at time level $n+1$, can be written in matrix form respectively, as:

$$(\mathbf{M}_u+\mathbf{A})\underline{\mathbf{u}}^{n+1} = \mathbf{C}\underline{\mathbf{p}}^{n+1} + \underline{\mathbf{s}}_u^{n+1}, \tag{11}$$

$$\mathbf{B}^{\mathrm{T}}\underline{\mathbf{u}}^{n+1} = \underline{\mathbf{s}}_p^{n+1}, \tag{12}$$

where $\underline{\mathbf{u}}^{n+1}$ and $\underline{\mathbf{p}}^{n+1}$ are the FEM solution fields for velocity and pressure, respectively, and $\underline{\mathbf{s}}_u^{n+1}$ and $\underline{\mathbf{s}}_p^{n+1}$ are discretised sources.

For generality, we have introduced a matrix $\mathbf{A}$, containing advection and diffusion contributions, which may be distributed and thus cannot be easily inverted. This allows the method to be applied to inertia-dominated or viscous-dominated flows without modification. Since the velocity is discontinuous between elements, the mass matrix $\mathbf{M}_u = \int_{V_E} \mathbf{Q}_i \mathbf{Q}_j dV$ is block-diagonal and thus can be easily inverted, each block being local to an element.

The solution method proceeds by first solving for an intermediate velocity $\underline{\mathbf{u}}_*^{n+1}$ using a guessed pressure $\underline{\mathbf{p}}_*^{n+1}$. On the first iteration within a time step one may use $\underline{\mathbf{p}}_*^{n+1} = \underline{\mathbf{p}}^{n}$. Eq. (11) becomes:

$$(\mathbf{M}_u + \mathbf{A})\underline{\mathbf{u}}_*^{n+1} = \mathbf{C}\underline{\mathbf{p}}_*^{n+1} + \underline{\mathbf{s}}_u^{n+1}. \tag{13}$$

The matrix equation for velocity to be satisfied is:

$$\mathbf{M}_u\underline{\mathbf{u}}^{n+1} + \mathbf{A}\underline{\mathbf{u}}_*^{n+1} = \mathbf{C}\underline{\mathbf{p}}^{n+1} + \underline{\mathbf{s}}_u^{n+1}. \tag{14}$$

Subtracting these two equations the velocity correction equation is obtained:

$$\mathbf{M}_u(\underline{\mathbf{u}}^{n+1} - \underline{\mathbf{u}}_*^{n+1}) = \mathbf{C}(\underline{\mathbf{p}}^{n+1} - \underline{\mathbf{p}}_*^{n+1}). \tag{15}$$

Multiplying this equation by $\mathbf{B}^{\mathrm{T}}\mathbf{M}_u^{-1}$ (see Eq. (12)) and using the global continuity equation to eliminate out $\underline{\mathbf{u}}^{n+1}$ one obtains the pressure correction equation:

$$\mathbf{B}^{\mathrm{T}}\mathbf{M}_u^{-1}\mathbf{C}(\underline{\mathbf{p}}^{n+1} - \underline{\mathbf{p}}_*^{n+1}) = -\mathbf{B}^{\mathrm{T}}\underline{\mathbf{u}}_*^{n+1} + \underline{\mathbf{s}}_p^{n+1}. \tag{16}$$

This equation is solved for $\underline{\mathbf{p}}^{n+1}$ and the velocity is corrected using Eq. (15).

This pressure equation is typically solved using the implementation of the GMRES Krylov subspace solver [25] in the PETSc framework [26]. The Boomer algebraic multi-grid preconditioner from the HYPRE library [27] is used to accelerate the process. The velocity solutions are calculated using GMRES [25] with SOR preconditioning.

It is worth mentioning that in the first time step after the mesh adaption, the interpolated velocity field on the new mesh is projected to a continuity-satisfied space, otherwise it will not satisfy the divergence free condition. Thus, the following calculation steps are performed: the interpolated velocity after a mesh adaptation is placed into the right hand side (rhs) of the pressure equation (Eq. (16)); this is then used to produce a pressure correction that is placed in the rhs of the velocity correction equation (Eq. (15)); these velocity and pressure corrections are then added to the interpolated velocity and pressures and we commence time stepping from these values. If this modification is not made, then after a mesh adaption, there is typically a spike in pressure and velocity magnitude in isolated regions. This can result in poor accuracy in these regions and this typically leads to the simulations becoming unstable.

2.2.5. Interface capturing method

The algebraic VOF method is employed here for the interface capturing. It is based on a compressive advection method, which uses a novel and mathematically rigorous non-linear Petrov–Galerkin method that attempts to keep interfaces between components sharp. The mass conservation for each components Eq. (2) is solved using a control volume and finite element formulation, involving a high-order accurate finite element method to obtain fluxes on the control volume boundaries, where these fluxes are subject to flux-limiting using a NVD approach [12] to obtain bounded and compressive solutions for the interface. More details can be found in [16].

2.2.6. Surface tension model

The surface tension force $\mathbf{F}_\sigma$ in Eq. (3) is obtained via the continuum surface force (CSF) method [20] as:

$$\mathbf{F}_\sigma = \sigma\kappa\tilde{\mathbf{n}}\delta, \tag{17}$$

where σ is the surface tension coefficient, κ is the interfacial curvature, $\tilde{\mathbf{n}}$ is the interface unit normal, δ is the Dirac delta function. Here, we use $\delta = |\nabla\alpha|$ and $\tilde{\mathbf{n}} = \frac{\nabla\alpha}{|\nabla\alpha|}$ to reformulate the CSF based on the component volume fraction as:

$$\mathbf{F}_\sigma = \sigma\kappa\nabla\alpha. \tag{18}$$

In order to evaluate the curvature κ, it is normally estimated as $\kappa = \nabla\cdot\tilde{\mathbf{n}}$ based on the volume fraction or height function technique [22] in conventional volume of fluid type approaches. However, as the volume fraction is a step function between zero and one and it is not easy to calculate a normal, sometimes it will produce spurious velocities even when using a smoothed volume fraction to calculate the curvature. The height function technique is a good alternative and higher order accurate for Cartesian structured grids, however it is difficult to reconstruct the height function for a fully unstructured mesh. Recently, a number of mesh-decouple height function methods have been developed. Owkes et al. [28] used interface normal aligned columns to get a better estimate of curvature for under-resolved interfaces and Ivey et al. [29] constructed height function structured stencils from an unstructured grid based on interpolation. Both methods use the geometric VOF schemes on a fixed mesh. In order to deal with algebraic VOF method on an adaptive mesh, two different curvature approximations are proposed here: distance function approach and diffused interface approach.

a. Distance function approach

In this approach, the curvature is estimated as:

$$\kappa = \nabla\cdot\tilde{\mathbf{n}}_{df}, \tag{19}$$

where $\tilde{\mathbf{n}}_{df} = \frac{\nabla\varphi}{|\nabla\varphi|}$ is the interface unit normal calculated from the signed distance function φ from the interface (similar to the level set function). Contrary to the standard CLSVOF method, only one function (volume fraction α) is advected to capture the interface here, and then a distance function φ is calculated based on the volume fraction α by the following three steps.

In the first step, the distance function is initialised as:

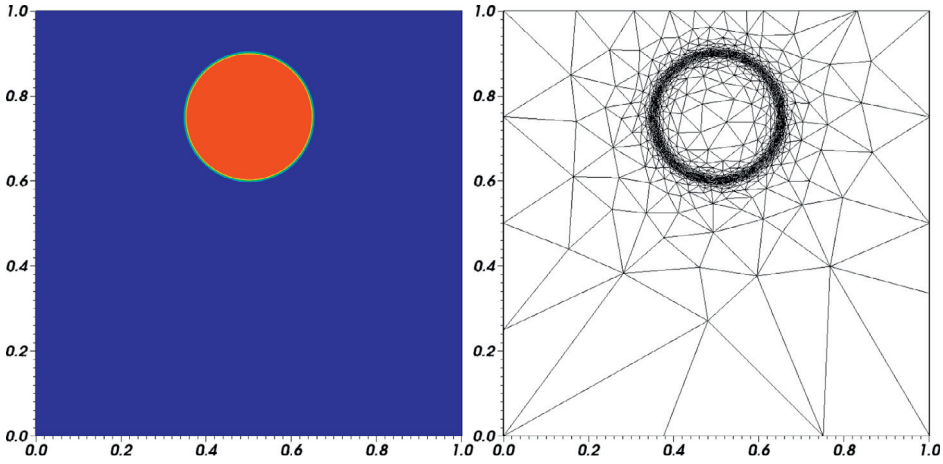
$$\varphi_0 = (2\alpha - 1)h_{\min}, \tag{20}$$

where $h_{\min}$ is the minimum mesh size around the interface. As we assume the contour of $\alpha = 0.5$ is the interface, it can be seen that Eq. (20) provides a good estimate for the initial distance function, where it is zero at the interface (required by the definition) and has a different sign on either side of the interface with a good guess for the distance in the vicinity of the interface.

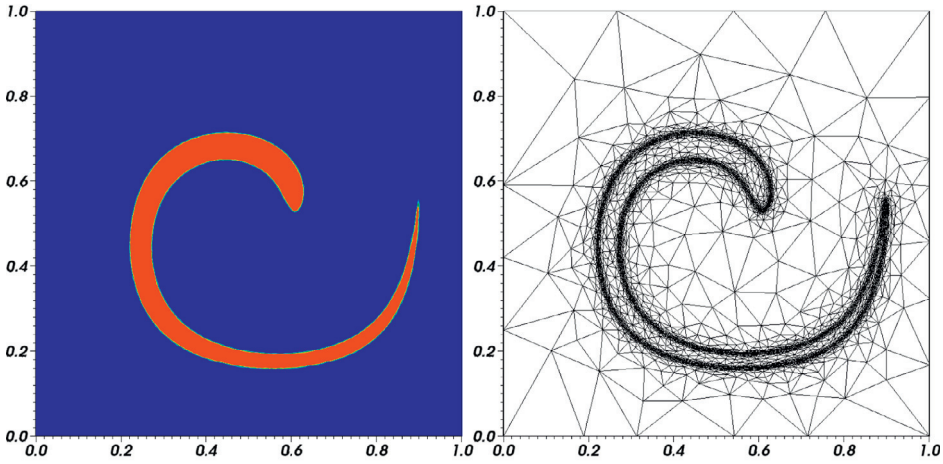
In the second step, we follow the re-initialisation process in the level set method to obtain the actual sign distance function φ by solving the following equation with the initial value φ_0:

$$\frac{\partial\varphi}{\partial\tau} = S(\varphi_0)(1 - |\varphi|), \tag{21}$$

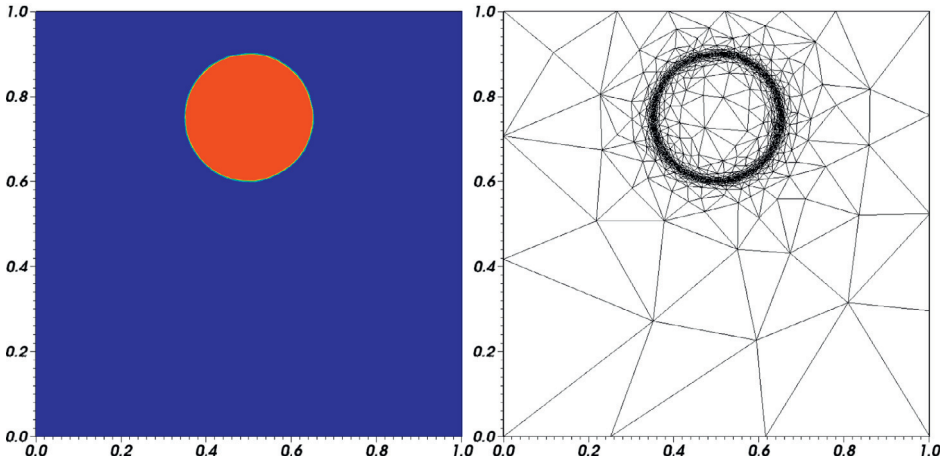
where $S(\varphi_0) = \varphi_0/\sqrt{\varphi_0^2 + \varepsilon}$ is the sign function and $\varepsilon = h_{\min}$ is used here and τ is an artificial time. We iterate on Eq. (21) no more than $h_s/\Delta\tau$ times in order to get the converged distance function solution for $|\nabla\varphi| = 1$, where $2h_s$ represent the transition region around the interface and $h_s = 1.5h_{\min}$ and $\Delta\tau = 0.1h_{\min}$ are used here. It is worth noting that a discontinuous Galerkin method for φ is used here during iteration in order to make it very accurate. In addition, we use the pressure basis functions as it is high order.



(a) $t = 0.0$



(b) $t = 2.0$



(c) $t = 4.0$

Fig. 2. Interface and mesh under the single-vortex shearing flow for $T = 4$ and $h_{\min} = 1/512$. The adaptive mesh provides a fine resolution equivalent to that of a 512 × 512 uniform mesh.

In the last step, the calculated discontinuous distance function (φ^{DG}) is projected to a continuous space (φ^{CG}) by a volume-weighted interpolation as:

$$\varphi^{\mathrm{CG}} = \frac{\sum_{i=1}^{N_{cv}} \varphi_i^{\mathrm{DG}} V_i}{\sum_{i=1}^{N_{cv}} V_i}, \tag{22}$$

where N_{cv} is the number of control volumes connected to the continuous function φ^{CG} and V_i is the volume of the control volume (see Fig. 1 for example). Then the continuous function φ^{CG} is used to calculate the gradient, which helps to accurately estimate the unit normal $\tilde{\mathbf{n}}_{df}$ and the curvature κ.

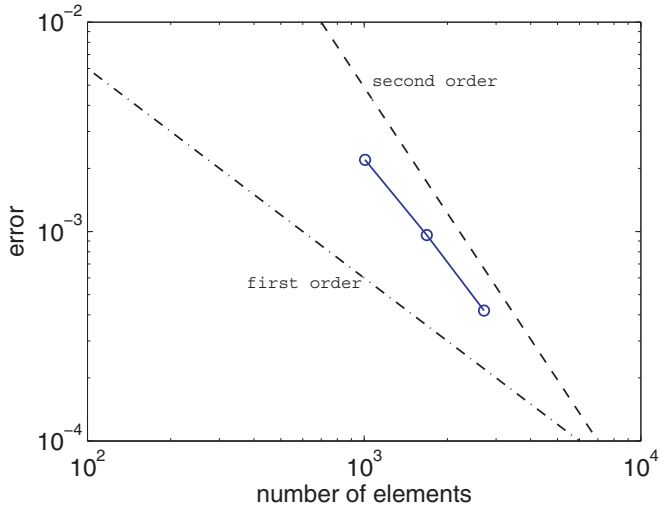


Fig. 3. Average error of the interfacial location as function of the number of elements. The order of convergence is about 2 and the first-order slope is shown as dash-dotted line and the second-order slope is shown as dashed line.

b. Diffused interface approach

In this approach, we use a diffused interface based on the original volume fraction α to estimate the curvature as:

$$\kappa = \nabla \cdot \tilde{\mathbf{n}}_{di}, \tag{23}$$

where $\tilde{\mathbf{n}}_{di} = \frac{\nabla\psi}{|\nabla\psi|}$ is the interface unit normal calculated from the diffused volume fraction ψ from the interface. In contrast to some smoothing techniques for the volume fraction to calculate the curvature, we obtain the diffused volume fraction by solving a diffusion equation as:

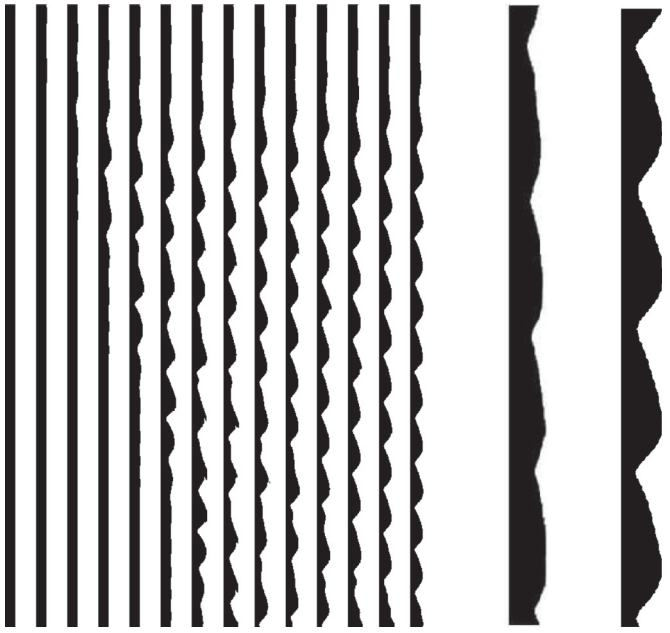
$$\frac{\partial\psi}{\partial\tau} = D\Delta\psi, \tag{24}$$

where τ is also an artificial time and D is an artificial diffusion coefficient. Here, we initialise the diffusion value based on the original volume fraction α as:

$$\psi_0 = \alpha h_{\min}, \tag{25}$$

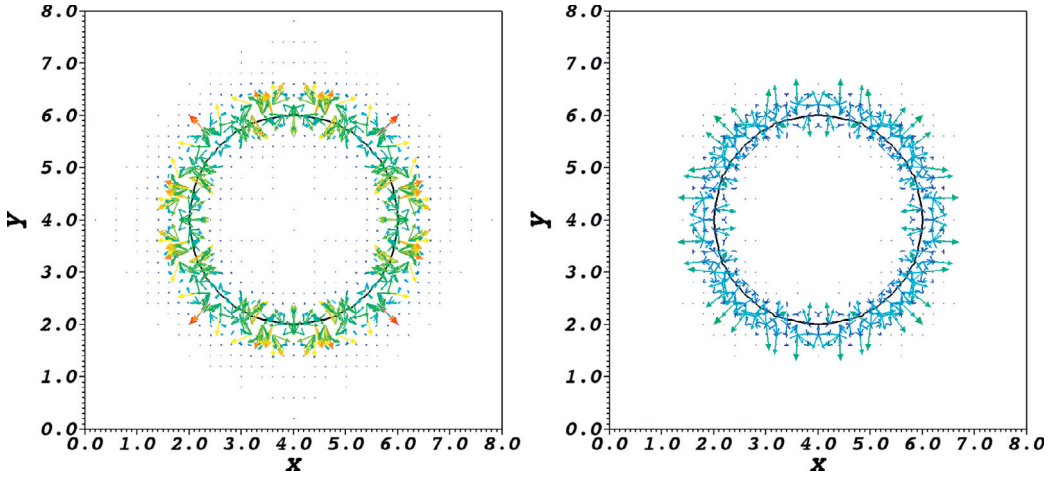
which is relevant to the dimension of the computational domain or size of the mesh. After that, we solve the Eq. (24) with a discontinuous Galerkin finite element method with $\Delta\tau = 1$ and $D = h_{\min}$ for a few iteration (normally less than 5 times as we only need the diffused value around the interface), in order to get a high-order accurate result. A diffused interface can be obtained with a smooth variation for the diffused volume fraction ψ, which provides an easy way to calculate the gradient of its value.

Finally, similar to the distance function approach, the calculated discontinuous diffused volume fraction is projected to a continuous space, which is used to approximate the unit normal $\tilde{\mathbf{n}}_{di}$ and the curvature κ.



(a) computed temporal evolution (b) comparison

Fig. 5. Falling liquid film for $Re = 20.1$ and $We = 33.5$ with forcing frequency 27 Hz: (a) numerical results for the evolution of the falling liquid film flow with 0.05 s interval (the film has been scaled 10 times in the x direction in order the see the wave) and (b) comparison of wave shape between experiment (left) and simulation (right).



(a) calculated distance function for κ (b) diffused interface for κ

Fig. 4. Numerical results for the velocity field with different curvature calculations after one time step with $\Delta t = 10^{-6}$ for the inviscid static drop in equilibrium for density ratio $\rho_1/\rho_2 = 10^3$ for the structured mesh. Both have the same scale and the solid line represents the interface.

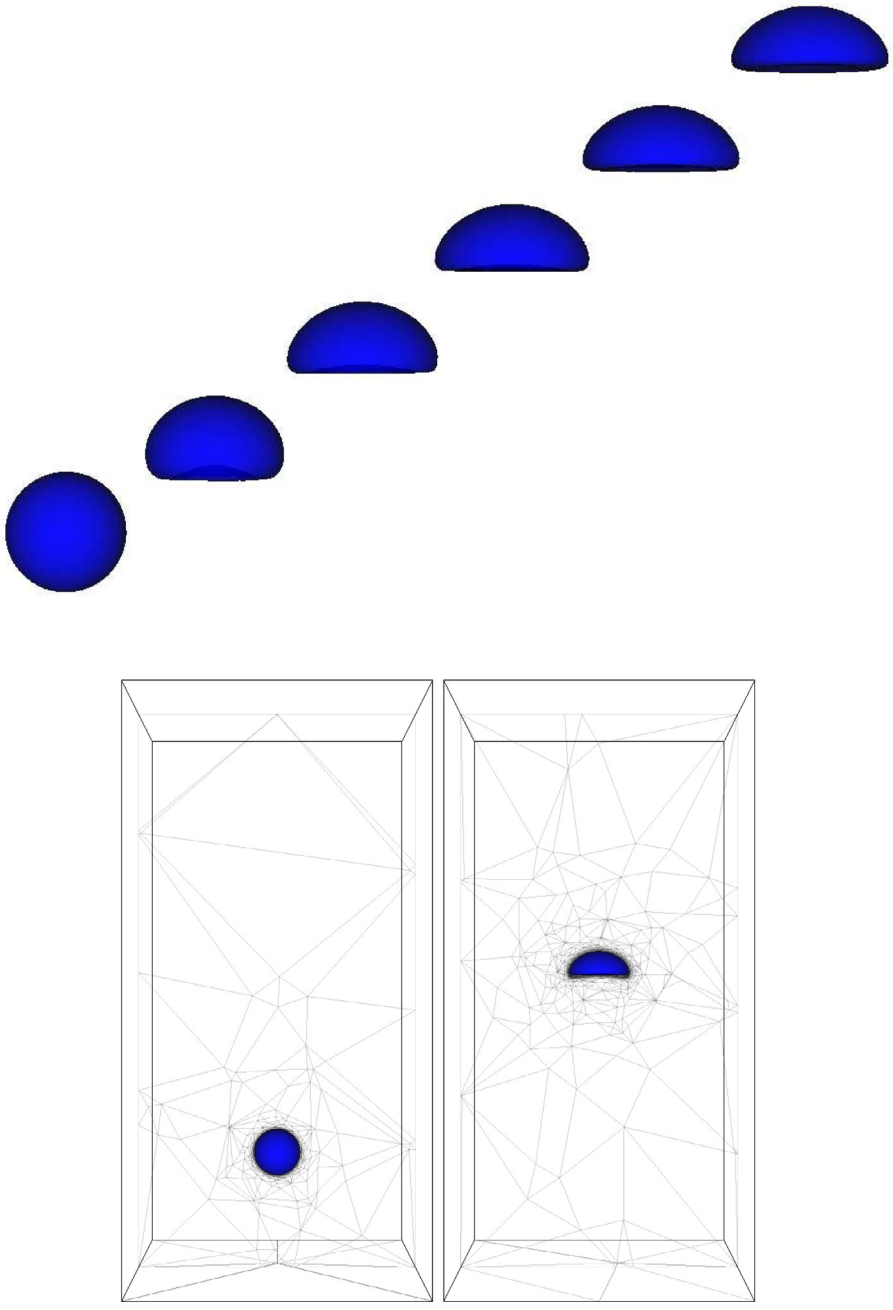


Fig. 6. Snapshots of three-dimensional numerical simulation of a single rising bubble. Top panel shows the zoom in interface from $t = 0$ s to $t = 0.25$ s with 0.05 s interval. Bottom panel shows the profile alongside with the unstructured mesh at the bottom and central slices for $t = 0$ s and $t = 0.25$ s.

2.2.7. Mesh adaptivity algorithm

A common problem encountered by model users is that the computational mesh used in a numerical simulation has to be generated *a priori* to the solution procedure. It is therefore difficult to resolve adequately the local physical features at a first attempt, and the mesh often needs to be adapted to enable the solution procedure to satisfy resolution requirements. Importantly, this also allows for a reduction in computational effort which is crucial for complex applications.

The present model adapts the mesh to the solution without sacrificing the integrity of the boundary (geometry), or internal boundaries (regions) of the domain. It circumvents the complexities of boundary-conforming Delaunay methods by operating on the existing mesh. The error measure employed is based on the curvature of the solution and provides a directional measure. The objective is to obtain a mesh which has a uniform interpolation error in any direction. This is accomplished with use of a metric which is related to the Hessian of the solution field. Appropriate scaling of the metric enables the resolution of multi-scale phenomena as encountered in multiphase flows. The resulting metric is used to calculate element size and shape. The mesh optimisation method is based on a series of mesh connectivity and node position searches of the landscape, defining mesh quality which is gauged by a functional. The mesh modification thus fits the solution field(s) in an optimal manner. The anisotropic mesh adaption technique developed by [30] is used here. In this paper, the pressure and volume fraction projection use the consistent interpolation, while the

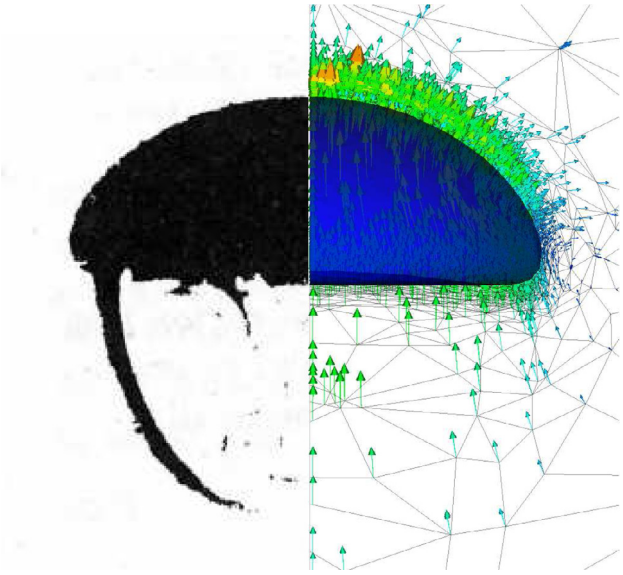


Fig. 7. Comparison of the bubble shape between experimental measurement [34] (left) and numerical simulations (right) together with the velocity field and mesh on the central plane.

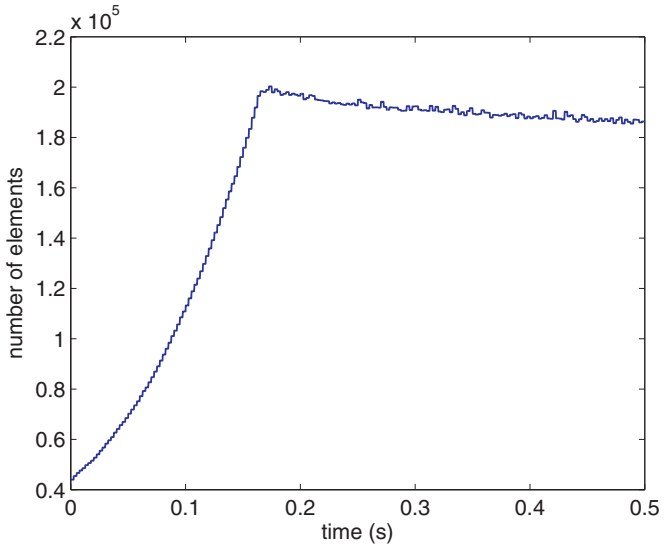


Fig. 8. Number of total elements for the three-dimensional numerical simulation of a single rising bubble with an adaptive unstructured mesh.

velocity projection uses a conservative Galerkin interpolation technique [31].

3. Numerical examples

3.1. Single vortex

Before we apply our method to coupled fluid flow problems, we first test our control volume-based interface-capturing method for a pure advection problem where the fluid interface moves under a prescribed velocity field. The single vortex problem [4], which is widely used as a benchmark test, is considered here using P_2 elements. The forward Euler time-stepping scheme is used here for the advection case, whereas the Crank–Nicolson and backward Euler schemes are employed for the volume fraction and momentum equations in the rest of the paper. A circle (radius 0.15) is initially centered at (0.50, 0.75) in a unit square computational domain. The velocity field is defined by the stream function Ψ as:

$$\Psi = \frac{1}{\pi}\sin^2(\pi x)\cos^2(\pi y)\cos\left(\frac{\pi t}{T}\right), \quad u = -\frac{\partial \Psi}{\partial y}, \quad v = \frac{\partial \Psi}{\partial x}, \tag{26}$$

where u and v are the horizontal and vertical components of the velocity field, respectively.

The initial interface shape is deformed by the velocity field and return to its initial state at $t = T$, where $T = 4$ is used in the simulation. In order to avoid the influence of time step on the results, $\Delta t = 2.5 \times 10^{-4}$ is used for three different computations with minimum mesh size $h_{\min}$ as 1/128, 1/256 and 1/512. Fig. 2 shows the interface shape alongside the mesh during the simulation for the computation with $h_{\min} = 1/512$. It can be seen that the circle is stretched from $t = 0$ under the specified velocity field until $t = 2$ reaching its maximum deformation. At this stage, it is a spiral shape with very elongated filament which is very thin at the tail. The interface has been efficiently captured in the computation by using the adaptive unstructured mesh, which provides fine resolution equivalent to that of a 512 × 512 uniform mesh. After $t = 2$, the velocity field is reversed and the interface shape is returned to its initial shape which is well captured during the computation. This test demonstrates the power of the adaptive mesh approach, which can refine the mesh in the vicinity of the interface or an area of interest, and reduce computational effort without sacrificing accuracy.

Table 1
Error in velocity and pressure drop after one time step with $\Delta t = 10^{-6}$ for the inviscid static drop in equilibrium with three different density ratios when the exact curvature is specified for the structured mesh.

Method	ρ_1/ρ_2	$\lvert\mathbf{u}\rvert_{\max}$	$E(\Delta P)_{\text{total}}$	$E(\Delta P)_{\text{partial}}$
Present	1	7.77×10^{-19}	6.58×10^{-2}	3.35×10^{-17}
	10^3	9.09×10^{-19}	6.58×10^{-2}	3.40×10^{-20}
	10^5	9.11×10^{-19}	6.58×10^{-2}	6.78×10^{-21}
CSF model [22]	1	1.25×10^{-18}	2.89×10^{-2}	2.73×10^{-15}
	10^3	4.97×10^{-18}	2.89×10^{-2}	3.89×10^{-16}
	10^5	5.70×10^{-18}	2.89×10^{-2}	1.95×10^{-16}

As the location of the interface is known (which is represented as the contour of the volume fraction at $\alpha = 0.5$), the deviation of the interface position after one rotation can be calculated as:

$$\text{error} = \frac{1}{N}\sum_{1}^{N} \left|\sqrt{(x-0.5)^2 + (y-0.75)^2} - 0.15\right|, \tag{27}$$

where N is the number of points along the interface. Fig. 3 shows the convergence for the computations with three different simulations. It can be seen that the present method is close to second order accurate, which is consistent with the quadratic polynomial function used for the P_2 finite element type.

3.2. Static drop in equilibrium

In order to validate the proposed framework for surface tension, we consider the inviscid static drop in equilibrium problem [22], which is a benchmark case for testing the surface tension method without viscosity and gravity. We follow the same computational setup as in [22]. A drop (radius $R = 2$) is positioned at the centre of the computational domain $[0, 8] \times [0, 8]$, the surface tension coefficient $\sigma = 73$, the density inside the drop is $\rho_1 = 1$ and outside density ρ_2 is varied from 0 to 1. A uniform computational mesh with 40 layers in each direction is used for the computations with the P_1DG-P_2 element pair.

In the first set of calculations, we test the coupling of surface tension force with the pressure gradient by specifying the exact curvature $\kappa_{\text{exact}} = 1/R = 0.5$. Table 1 shows the comparison for the maximum velocity and pressure jump errors after one time step

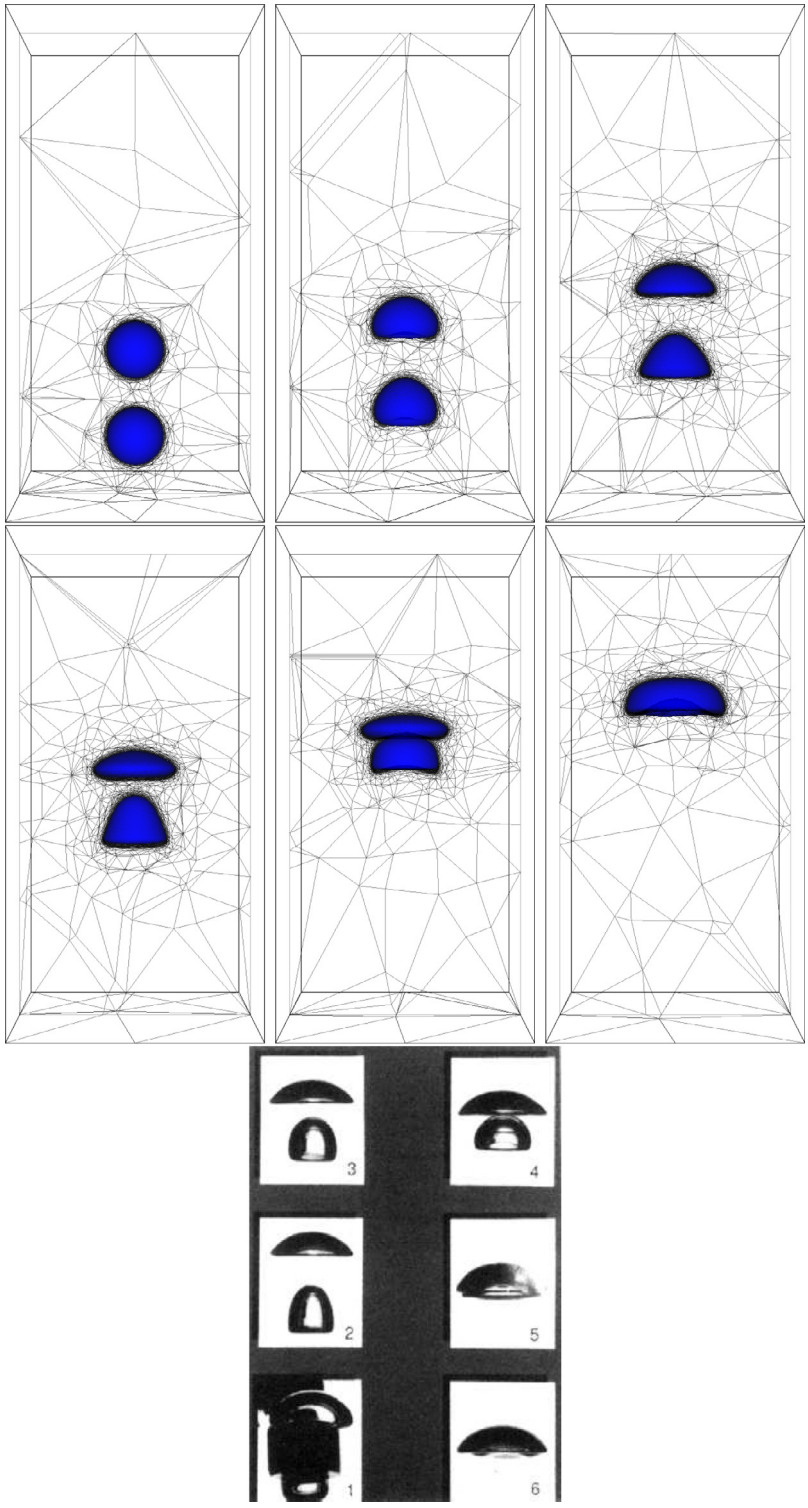


Fig. 9. Comparison of the evolution of the bubble shape between the numerical simulations (top and middle) and experimental measurement [37] (bottom) for the co-axial coalescence of two bubbles initially at [0, 2.5D, 0] and [0, D, 0] in a quiescent liquid. The numerical simulation shows the interface shape alongside with the unstructured mesh at the bottom and central slices from $t = 0$ s to $t = 0.15$ s with 0.03 s interval and the time difference between subsequent photographs is 0.03 s in the experiments.

for three different density ratios, and also with the results obtained for the structured Cartesian mesh in [22]. These errors are defined as:

$$|\mathbf{u}|_{\max} = \max(|\mathbf{u}|), \tag{28}$$

$$E(\Delta P)_{\mathrm{w}} = \frac{|\Delta P_{\mathrm{w}} - \Delta P_{\mathrm{exact}}|}{\Delta P_{\mathrm{exact}}}, \tag{29}$$

where $\Delta P_{\mathrm{exact}} = \sigma\kappa = 36.5$, and w denotes different evaluation ways by using pressure points in the areas of $r < R$ and $r > R$ (total) and the areas of $r < R/2$ and $r > 3R/2$ (partial), where 'total' considers the whole region inside and outside the drop and 'partial' considers some parts of the region by avoiding the transition zone. It can be seen from Table 1 that the spurious currents are very small, related to the machine accuracy. For the pressure drop, the total error $E(\Delta P)_{\mathrm{total}}$ is of the order of 10^{-2}, and is inde-

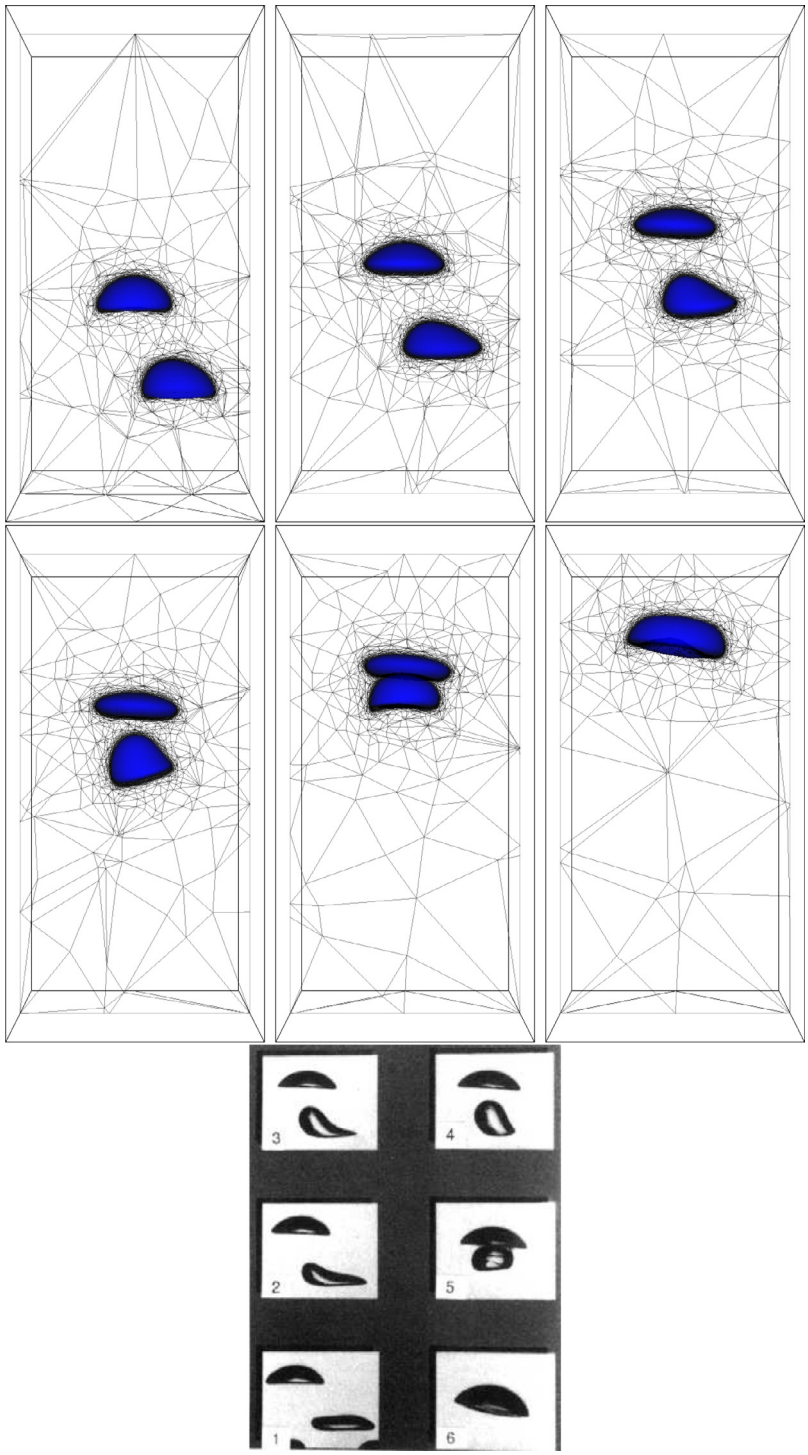


Fig. 10. Comparison of the evolution of the bubble shape between the numerical simulations (top and middle) and experimental measurement [37] (bottom) for the oblique coalescence of two bubbles initially at [0, 2.5D, 0] and [0.8D, D, 0] in a quiescent liquid. The numerical simulation shows the interface shape alongside with the unstructured mesh at the bottom and central slices from $t = 0.03$ s to $t = 0.18$ s with 0.03 s interval and the time difference between subsequent photographs is 0.03 s in the experiments.

pendent of the density ratio. The partial error $E(\Delta P)_{\text{partial}}$ is much smaller because the error measurement does not include the transition region. Table 1 also shows that the present surface tension model, which can be extended for the fully unstructured mesh, is comparable to the balanced-force surface tension model for the structured Cartesian mesh.

In the second set of calculations, we test the effects of different techniques for curvature calculations for the spurious velocity and pressure drop for the same setup above for density ratio $\rho_1/\rho_2 = 10^3$. Table 2 shows the results for four different tech-

Table 2
Error in velocity and pressure drop after one time step with $\Delta t = 10^{-6}$ for the inviscid static drop in equilibrium for density ratio $\rho_1/\rho_2 = 10^3$ with different curvature calculation techniques for the structured mesh.

Method	$\lvert\mathbf{u}\rvert_{\max}$	$E(\Delta P)_{\text{total}}$	$E(\Delta P)_{\text{partial}}$
Exact κ	9.09×10^{-19}	6.58×10^{-2}	3.40×10^{-20}
Exact distance function for κ	2.39×10^{-5}	6.70×10^{-2}	3.62×10^{-3}
Calculated distance function for κ	6.22×10^{-5}	7.02×10^{-2}	3.80×10^{-3}
Diffused interface for κ	2.31×10^{-5}	6.61×10^{-2}	4.60×10^{-3}

Table 3
Error in velocity and pressure drop after one time step with $\Delta t = 10^{-6}$ for the inviscid static drop in equilibrium for density ratio $\rho_1/\rho_2 = 10^3$ with the distance function approach and diffused interface approach for the curvature calculation for the fully unstructured mesh.

Method	$\|\mathbf{u}\|_{\max}$	$E(\Delta P)_{\text{total}}$	$E(\Delta P)_{\text{partial}}$
Calculated distance function for κ	1.6×10^{-4}	6.52×10^{-2}	7.46×10^{-4}
Diffused interface for κ	1.5×10^{-4}	6.37×10^{-2}	2.5×10^{-3}

niques: the first is for the given exact curvature; the second is for the curvature calculated from the analytically given distance function (the distance function is given rather than solving from Eq. (21); the third and fourth are for the distance function approach and diffused interface approach for the curvature calculation proposed in this study, respectively. It can be seen from Table 2 that the two approaches for the curvature calculation are in the same order of accuracy, whereas the diffused interface approach performs slightly better than the distance function approach, and even better or close to the case when the distance function is specified. The velocity fields obtained from the two approaches for the curvature calculation are shown in Fig. 4.

In the third set of calculations, we test the distance function approach and diffused interface approach for the curvature calculation proposed in this study for the same setup above for density ratio $\rho_1/\rho_2 = 10^3$, but with a fully unstructured mesh with similar resolution compared to the structured mesh. Table 3 shows the results for the two approaches and it can be seen that similar results are obtained for the unstructured mesh. Compared to the results for the structured mesh, the pressure drop is in the same order whereas the spurious currents are larger due to the curvature approximation for the unstructured mesh.

As shown in Fig. 4 and Table 3, slightly smaller spurious velocities are observed when using the diffused interface approach to calculate κ, thus we used the diffused interface approach for the curvature calculation in the rest of the paper.

3.3. Falling liquid film

A 2D falling liquid film in contact with air is considered here to test the surface tension treatment in a fluid flow problem. We follow the same computational setup as in [32]. The computational domain dimensions are $[0, 600h_N] \times [-h_N, 3h_N]$, where h_N is the undisturbed film thickness (Nusselt height). The computational domain is discretised by 1500 × 25 layers, whereas the mesh is uniform in the x direction and mesh size is $\Delta y = 0.1h_N$ from $[-h_N, 0.5h_N]$ and gradually increases to the end in the y direction. A no-slip boundary condition is applied at both walls in the water and air side, and a zero-gradient boundary condition is applied at the outlet. At the inlet $x = 0$, a parabolic velocity profile is imposed for the liquid phase as:

$$u_{\text{in}}(y \leqslant 0) = \frac{3}{2}\left(1 + \frac{y}{h_N}\right)\left(1 - \frac{y}{h_N}\right)u_{\text{av}}, \tag{30}$$

where u_{av} is the average velocity containing the forcing perturbation:

$$u_{\text{av}} = u_N(1 + \xi \sin(2\pi F t))), \tag{31}$$

in which u_N is the Nusselt velocity, ξ is the disturbance magnitude and F is the forcing frequency. The velocity for the air phase at the inlet is set as:

$$u_{\text{in}}(y > 0) = \frac{3}{2}\left(1 + \frac{y}{3h_N}\right)\left(1 - \frac{y}{3h_N}\right)u_{\text{av}}. \tag{32}$$

The simulation is initialised with a flat film with a fully developed velocity field as prescribed at the inlet. The computation is carried out using the $P_1DG\text{-}P_2$ element pair. The liquid film flow is simulated for $Re = \frac{\rho u_N h_N}{\mu} = 20.1$ and $We = \frac{\rho u_N^2 h_N}{\sigma} = 33.5$ with the computational parameters shown in Table 4, which corresponds to the experimental case reported in [33].

Table 4
Computational parameters used for the falling liquid film simulation.

variables	value	units
Film thickness h_N	1.83×10^{-4}	m
Mean velocity u_N	0.10865	m/s
Forcing frequency F	27	Hz
Liquid density	1000	kg/m^3
Liquid viscosity	9.892×10^{-4}	kg/(m*s)
Air density	1.125	kg/m^3
Air viscosity	1.81×10^{-5}	kg/(m*s)
Surface tension	0.072	N/m

Fig. 5(a) shows numerical results for the evolution of the falling liquid film under the inlet forcing perturbation. It can be seen that small waves grow in time and become largest in the development region. The waves are transported downstream from the inlet to the outlet. During the evolution, large waves travel faster to catch up with smaller waves due to gravity acceleration, forming even larger waves. The biggest wave merges with the front wave until it leaves the outlet. After that, nearly steady-state finite-amplitude travelling-waves are observed behind, which have nearly the same shape and amplitude, travelling with the same speed. The calculated steady-state wave shape is compared with the experimental measurement in Fig. 5(b), which shows they are very similar. The computed wave speed in the simulation is 0.213 m/s, which also agrees well with the measured wave speed 0.217 m/s in the experiment [33].

3.4. Single rising bubble

To validate the numerical framework for surface tension model with mesh adaptivity, we conduct three-dimensional (3D) numerical simulations of a single rising bubble and compare results with the experiments of [34] (case A in Table 1 in [34]), which have been used to validate two-phase flow codes [21,23,35,36]. In the simulation for the bubble with diameter D, we used a relative large computational domain $6D \times 12D \times 6D$ to avoid boundary effects whereas the centre of bubble is initialised at $y = 2.5D$. The computational parameters have been shown in [34] and in this case, the Reynolds number ($Re = \frac{\rho V R}{\mu}$) based on the liquid density, rising velocity V and effective radius R is 9.7, the Eotvos number ($Eo = \frac{g\Delta\rho D^2}{\sigma}$) is 38.9 and the Morton number ($M = \frac{g\mu^4\Delta\rho}{\rho^2\sigma^3}$) is 6.4×10^{-2}. The calculation is carried out with the $P_1DG\text{-}P_1$ element pair with the minimum length $h_{\min} = 0.01D$ being used in the test, which provides a fine resolution equivalent to that of a 600 × 1200 × 600 uniform mesh.

Fig. 6 shows a number of snapshots for the numerical results of the 3D rising bubble. It can be seen that the initially spherical bubble rises under the action of buoyancy, and undergoes deformation, resulting in the formation of the well-known cap-shaped bubbles. Fig. 6 also shows that the bubble shape has been well captured by refining the unstructured mesh in the vicinity of the deforming bubble preferentially. In contrast, the regions upstream and downstream of the bubble, which require a lower degree of resolution, have coarser elements, in order to maximise computational efficiency. At later stage, the bubble reaches a stable shape with a constant rising speed under the balance of surface tension, inertial force and the viscous force. Fig. 7 shows the comparison of the steady bubble shape between the experimental measurement and numerical simulations, which is in a good agreement. In addition, the predicted velocity field and the unstructured mesh along the central plane are also shown in Fig. 7.

In order to demonstrate the efficiency of the adaptive unstructured mesh simulation, Fig. 8 shows the time history of the total

number of elements during the simulations. Initially, only a few elements are used to resolve the bubble shape. The total number of elements gradually increases as finer meshes are placed to resolve the bubble deformation, and reaches a certain level when the bubble becomes a stable shape. Compared to the effective mesh size in a uniform Cartesian grid (a 600 × 1200 × 600 uniform mesh), the maximum total number of elements is only about 0.05% of the fixed mesh case, which can reduce the computational efforts without sacrificing accuracy.

3.5. Coalescence of two bubbles

Here we consider merging of two bubbles with co-axial and oblique coalescence. Two spherical gas bubbles with diameter $D = 0.01$ m are initially in a quiescent liquid in a computational domain $[-2D, 2D] \times [0, 8D] \times [-2D, 2D]$. For the co-axial case, the centre of the upper bubble is at $[0, 2.5D, 0]$ and the centre of the lower bubble is at $[0, D, 0]$; whereas for the oblique case, the centre of the lower bubble is shifted to $[0.8D, D, 0]$. The density and viscosity ratio (gas to liquid) is 0.001 and 0.01, respectively. For this problem, the Eotvos number ($Eo = \frac{g\Delta\rho D^2}{\sigma}$) is 16 and the Morton number ($M = \frac{g\mu^4\Delta\rho}{\rho^2\sigma^3}$) is 2×10^{-4} based on the liquid. In the simulations, an adaptive unstructured mesh with a minimum edge length $h_{\min} = 0.01D$ is used here with the P_1DG-P_1 element pair.

Fig. 9 shows the computed evolution of the bubble shape along with the adapted mesh and corresponding experimental photographs [37] for the co-axial coalescence case. It can be seen that compared to the leading bubble, the evolution of the tailing bubble is totally different, which is in the wake region of the leading bubble. The tailing bubble catches the leading bubble later on to form one single bubble. The computed evolution of the bubble shape is well captured by the adaptive unstructured mesh, which is in a good agreement with the experimental observations of [37].

For the oblique coalescence case, Fig. 10 shows the computed evolution of the bubble shape along with the adapted mesh and corresponding experimental photographs [37]. It can be seen that the dynamics are similar to the previous case and again good agreement between the computed and measured results is obtained here. It is worth noting that the flow field is clearly asymmetrical and three-dimensional, which cannot be captured in an axisymmetrical calculation.

3.6. Milkcrown problem

Finally, we consider the so-called 'milkcrown' problem which involves complex interface topology change including coalescence and breakup. Some researchers have studied the milkcrown problem [38,39] in 3D on a fixed mesh, here we simulate this problem with an adaptive mesh using the P_1DG-P_2 element pair in order to demonstrate the capability of the present framework. We follow the same setup as in [38] and [39], where a droplet (with diameter $D = 0.0053$ m) falls down with a speed of 2 m/s into thin film of the same liquid with 0.001 m depth. The density of the liquid and air is 1000 kg/m^3 and 1.25 kg/m^3, respectively. The viscosity of the liquid and air is 1.7×10^{-3} Pa s and 1.0×10^{-6} Pa s, respectively. The surface tension coefficient is 0.05 N/m and the acceleration due to gravity is 9.8 m/s^2. For this case, the Weber number for the liquid is $We = 426$. We used an adaptive unstructured mesh with the minimum length $h_{\min} \approx D/33$, which has the same resolution as mentioned in [38] and [39], but with much less number of degree of freedoms due to the adaptive mesh.

Fig. 11 shows snapshots of the interface shape along with the adaptive mesh used in the simulation. Similar results have been obtained in the present study compared to other results on a fixed mesh [38,39]. It can be seen that a liquid rim is generated after the droplet impact, which consequently breaks into smaller droplets. Fine mesh resolution is placed in the vicinity of the interface whereas the mesh coarsens away from the interface. Fig. 12 shows the time history of the total number of elements during the simulations and compared with the number for a fixed Cartesian grid with the equivalent resolution. It can be seen that the total number of elements gradually increases in order to capture some smaller structures in the flow and the maximum number is only about 13% of the fixed mesh case, showing the efficiency of the present method.

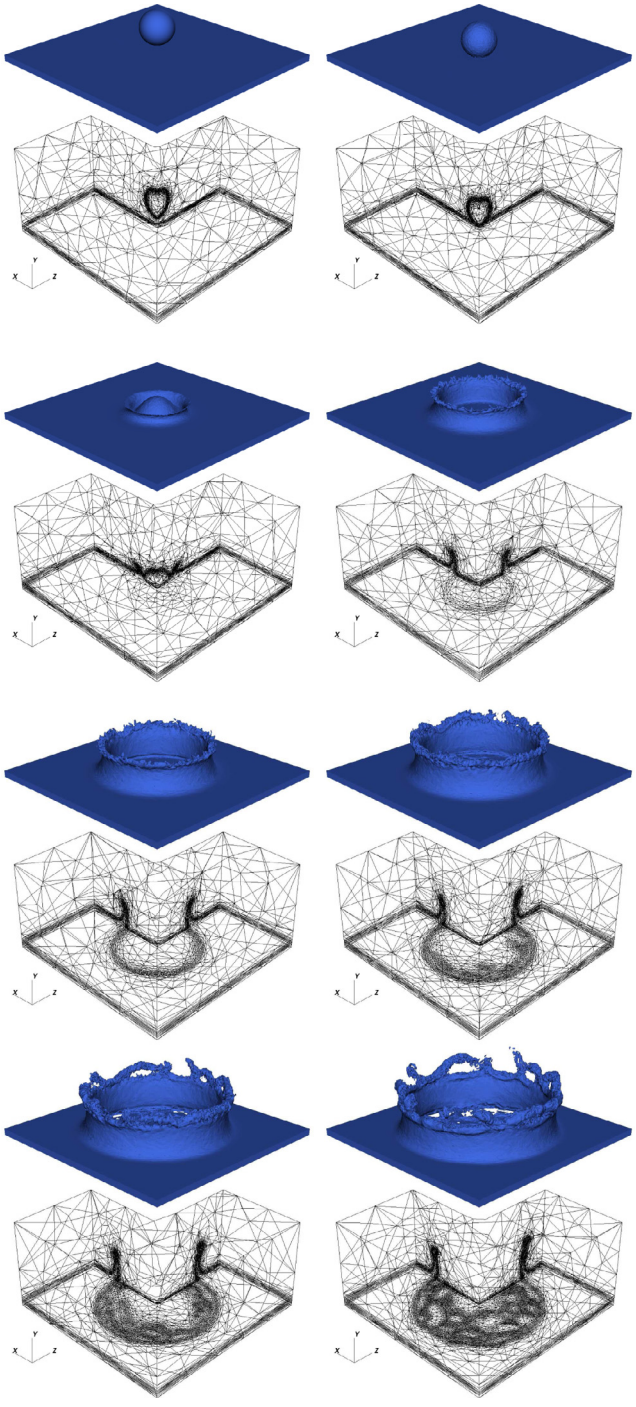


Fig. 11. Snapshots of numerical simulation of a droplet impact on a thin liquid layer. The droplet diameter is 0.0053 m and depth of the thin film is 0.001 m. The droplet initial velocity is 2 m/s which corresponds to a Weber number of 426. The predicted interface along with the adaptive unstructured mesh at three slices are shown at different times from the top left to bottom right at $t = 0.0, 0.001, 0.0025, 0.005, 0.0075, 0.01, 0.0125, 0.015$ s.

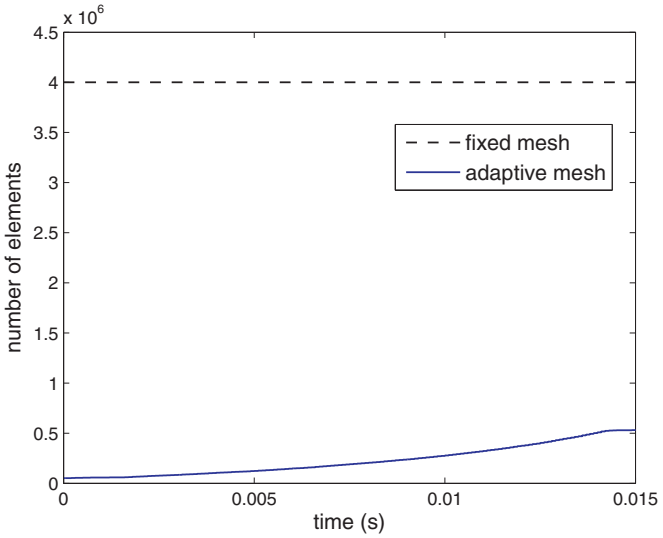


Fig. 12. Numbers of total elements for the numerical simulation of a droplet impact on a thin liquid layer with a fixed and adaptive, unstructured mesh.

4. Conclusions

In this paper a new balanced-force control volume finite element method with adaptive anisotropic unstructured meshes has been presented for interfacial flows with surface tension. A force-balanced algorithm for the surface tension implementation has been presented, with two different approaches to accurately estimate the curvature for a fully unstructured mesh, minimising the spurious velocities often found in such flow simulations. The numerical framework consists of a control volume and finite element formulation, a volume of fluid type method for the interface capturing based on a compressive advection method and an anisotropic adaptive mesh algorithm, which can modify and adapt unstructured meshes to better represent the underlying physics of interfacial problems and reduce computational effort without sacrificing accuracy. The numerical framework has been validated with several benchmark problems for interface advection, surface tension for equilibrium droplet, and dynamic fluid flow problems (fluid films, bubbles and droplets) in 2D and 3D.

The results presented here established with sufficient confidence that this method can be used to successfully model multiphase flows in a wide range of applications. This approach has the potential to be used for an arbitrary number of components although that has not been demonstrated here. Future work will include parallel computing and surfactant modelling.

Acknowledgements

We would like to thank the EPSRC MEMPHIS multiphase Programme Grant (number EP/K003976/1) for helping to fund this work. Dr D. Pavlidis would like to acknowledge the support from the EPSRC UK-Japan Civil Nuclear Energy project (EP/M012794/1) and H2020 IVMR. Funding for Dr P. Salinas from ExxonMobil is gratefully acknowledged.

References

[1] Scardovelli R, Zaleski S. Direct numerical simulation of free-surface and interfacial flow. Ann Rev Fluid Mech 1999;31:567–603.
[2] Harlow FH, Welch JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. Phys Fluids 1965;8:2182–9.
[3] Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. J Comput Phys 1981;39:201–25.
[4] Rider WJ, Kothe B. Reconstructing volume tracking. J Comput Phys 1998;141:112–52.
[5] Unverdi SO, Tryggvason G. A front-tracking method for viscous, incompressible, multi-fluid flows. J Comput Phys 1992;100:25–37.
[6] Osher SJ, Sethian JA. Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations. J Comput Phys 1988;79:12–49.
[7] Sethian JA, Smereka P. Level set methods for fluid interfaces. Ann Rev Fluid Mech 2003;35:341–72.
[8] Anderson DM, McFadden GB, Wheeler AA. Diffuse-interface methods in fluid mechanics. Ann Rev Fluid Mech 1998;30:139–65.
[9] Monaghan JJ. Smoothed particle hydrodynamics. Ann Rev Astronom Astrophys 1992;30:543–74.
[10] Rudman M. Volume-tracking methods for interfacial flow calculations. Int J Numer Methods Fluids 1997;24:671–91.
[11] Ubbink O, Issa RI. A method for capturing sharp fluid interfaces on arbitrary meshes. J Comput Phys 1999;153(1):26–50.
[12] Leonard BP. The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection 1991;88:17–74.
[13] Popinet S. An accurate adaptive solver for surface-tension-driven interfacial flows. J Comput Phys 2009;228:5838–66.
[14] Ceniceros HD, Roma AM, Silveira-Neto A, Villar MM. A robust, fully adaptive hybrid level-set/front-tracking method for two-phase flows with an accurate surface tension computation. Commun Comput Phys 2010;8:51–94.
[15] Zheng X, Lowengrub J, Anderson A. Adaptive unstructured volume remeshing ii: Application to two- and three-dimensional level-set simulations of multiphase flow. J Comput Phys 2005;208:626–50.
[16] Pavlidis D, Gomes JLMA, Xie Z, Percival JR, Pain CC, Matar OK. Compressive advection and multi-component methods for interface-capturing. Int J Numer Methods Fluids 2016;80:256–82.
[17] Xie Z, Pavlidis D, Percival JR, Gomes JLMA, Pain CC, Matar OK. Adaptive unstructured mesh modeling of multiphase flows. Int J Multiph Flow 2014;67:104–10.
[18] Pavlidis D, Xie Z, Percival JR, Gomes JLMA, Pain CC, Matar OK. Two- and three--phase horizontal slug flow modelling using an interface-capturing compositional approach. International Journal of Multiphase Flow. 2014;67:85–91.
[19] Piggott MD, Farrell PE, Wilson CR, Gorman GJ, Pain CC. Anisotropic mesh adaptivity for multi-scale ocean modelling. Philosoph Trans R Soc A 2009;367:4591–611.
[20] Brackbill J, Kothe D, Zemach C. A continuum method for modeling surface tension. J Comput Phys 1992;100:335–54.
[21] Sussman M, Puckett E. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. J Comput Phys 2000;162:301–37.
[22] Francois MM, Cummins SJ, Dendy ED, Kothe DB, Sicilian JM, Williams MW. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. J Comput Phys 2006;213:141–73.
[23] Yokoi K. A practical numerical framework for free surface flows based on clsvof method, multi-moment methods and density-scaled csf model: numerical simulations of droplet splashing. J Comput Phys 2013;232:252–71.
[24] Hirsch C. Numerical computation of internal and external flows - Volume 2: Computational methods for inviscid and viscous flows. John Wiley; 1990.
[25] Saad Y, Schultz MH. Gmres: a generalized minimal residual algorithm for solving non-symmetric linear systems. SIAM J Sci Stat Comput 1986;7:836–69.
[26] Balay S, Gropp WD, McInnes LC, Smith BF. Efficient management of parallelism in object oriented numerical software libraries. In: Arge E, Bruaset AM, Langtangen HP, editors. Modern Software Tools in Scientific Computing. Birkhäuser Press; 1997. p. 163–202.
[27] Henson VE, Yang UM. BoomerAMG: A parallel algebraic multigrid solver and preconditioner. Appl Numer Math 2002;41(1):155–77.
[28] Owkes M, Desjardins O. A mesh-decoupled height function method for computing interface curvature. J Comput Phys 2015;281:285–300.
[29] Ivey CB, Moin P. Accurate interface normal and curvature estimates on three--dimensional unstructured non-convex polyhedral meshes. J Comput Phys 2015;300:365–86.
[30] Pain CC, Umpleby AP, de Oliveira CRE, Goddard AJH. Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations. Comput Methods Appl Mech Eng 2001;190:3771–96.
[31] Farrell PE, Maddison JR. Conservative interpolation between volume meshes by local galerkin projection. Comput Methods Appl Mech Eng 2011;200:89–100.
[32] Gao D, Morley N, Dhir V. Numerical simulation of wavy falling film flow using vof method. J Comput Phys 2003;192:624–42.
[33] P. Kapitza, S. Kapitza, Collected Papers of P.L. Kapitza (vol.II), The Macmillan Company.
[34] Hnat J, Buckmaster J. Spherical cap bubbles and skirt formation. Physics of Fluids 1976;19:182–94.
[35] Gueyffier D, Li J, Nadim A, Scardovelli S, Zaleski S. Volume of fluid interface tracking with smoothed surface stress methods for three-dimensional flows. Journal of Computational Physics 1999;152:423–56.
[36] Xiao F, Honma Y, Kono T. A simple algebraic interface capturing scheme using hyperbolic tangent function. International Journal of Numerical Methods in Fluids 2005;48:1023–40.
[37] G. Brereton, D. Korotney, Dynamics of bubbles and vortices near a free surface, ASME, New York.
[38] Yokoi K. A numerical method for free-surface flows and its application to droplet impact on a thin liquid layer. J Sci Comput 2008;35:372–96.
[39] Chiu P, Lin Y. A conservative phase field method for solving incompressible two-phase flows. J Comput Phys 2011;230:185–204.